

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №2**  
**по дисциплине «Искусственные нейронные сети»**

Студент гр. 7381

\_\_\_\_\_

Лукашев Р.С.

Преподаватель

\_\_\_\_\_

Жукова Н.А.

Санкт-Петербург

2020

### **Цель работы.**

Реализовать классификацию между камнями (R) и металлическими цилиндрами (M) на основе данных об отражении сигналов радара от поверхностей. 60 входных значений показывают силу отражаемого сигнала под определенным углом.

### **Порядок выполнения работы.**

- Ознакомиться с задачей бинарной классификации
- Загрузить данные
- Создать модель ИНС в tf.Keras
- Настроить параметры обучения
- Обучить и оценить модель
- Изменить модель и провести сравнение. Объяснить результаты.

### **Требования.**

1. Изучить влияние кол-ва нейронов на слое на результат обучения модели.
2. Изучить влияние кол-ва слоев на результат обучения модели
3. Построить графики ошибки и точности в ходе обучения
4. Провести сравнение полученных сетей, объяснить результат

### **Ход работы.**

В соответствии с методическими указаниями была создана и натренирована нейронная сеть, решающая задачу бинарной классификации. В сети всего два слоя – входной и выходной, сеть тренировалась на 100 эпохах, размер пакета - 10. Точность и потери сети показаны на рисунке 1.

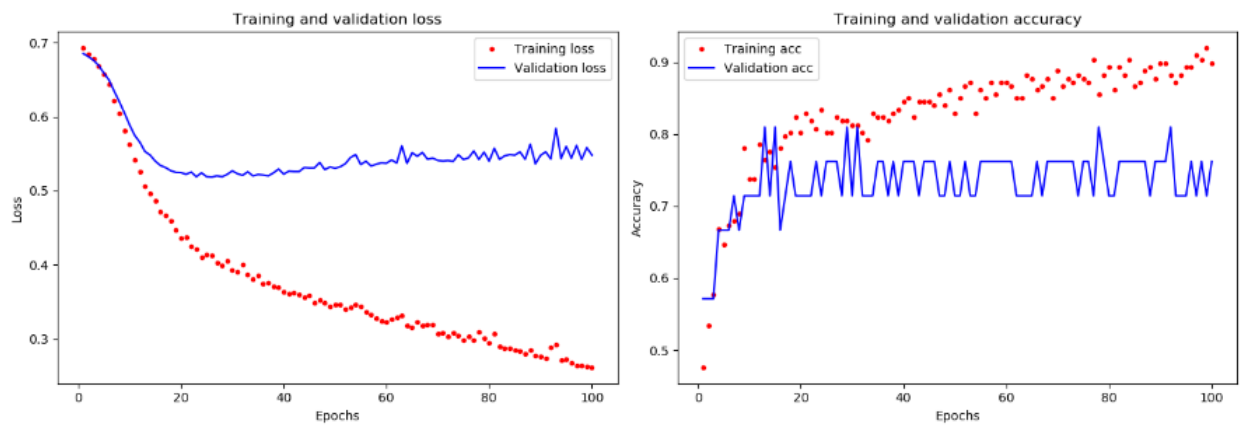


Рисунок 1 – точность и потери исходной модели.

Далее размер входного слоя был уменьшен вдвое. Результат тренировки показан на рисунке 2.

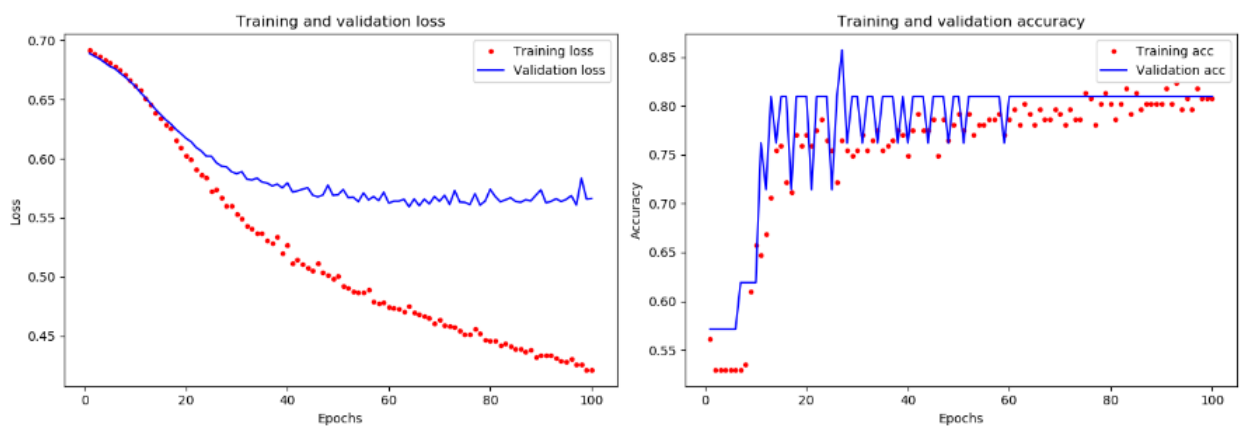


Рисунок 2 – точность и потери сети, в которой размер входного слоя уменьшен вдвое.

Далее к исходной модели был добавлен дополнительный слой на 15 нейронов. Результат тренировки показан на рисунке 3.

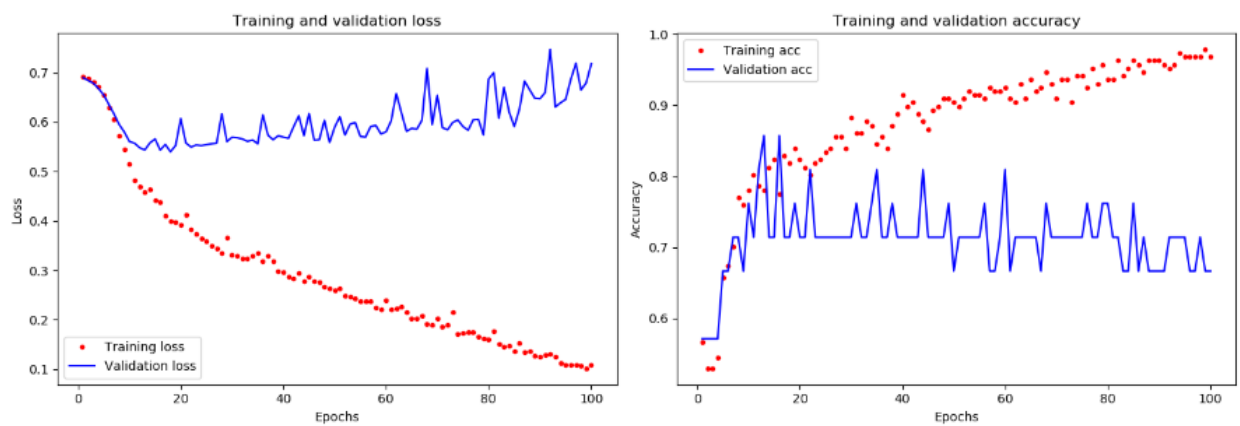


Рисунок 3 – точность и потери сети с дополнительным слоем.

Как можно заметить, в данном случае при увеличении количества слоев сеть заметно переобучается, т.е. ее точность на тестовых данных становится хуже, а на тренировочных лучше. При уменьшении количества нейронов на входном слое сеть обучается лучше, что говорит о том, что в входных данных присутствует некоторая избыточность, привносящая помехи, снижающие обобщающую способность сети.

### Выводы.

В ходе выполнения лабораторной работы были изучены различные архитектуры искусственных нейронных сетей, проведено обучение при различных параметрах, построены графики ошибок и точности в ходе обучения. Также было изучено влияние количества нейронов, избыточности входных данных, и количества слоев на результат обучения сети. При сверх меры большом количестве слоев для конкретных данных может происходить переобучение сети, т.е. сеть теряет способность к обобщению, но хороша в запоминании.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД

```
import numpy as np
import pandas
from keras.layers import Dense
from keras.models import Sequential
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder

dataframe = pandas.read_csv("sonar.csv", header=None)
dataset = dataframe.values
np.random.shuffle(dataset)

X = dataset[:, 0:60].astype(float)
Y = dataset[:, 60]

encoder = LabelEncoder()
encoder.fit(Y)
encoded_Y = encoder.transform(Y)
print(encoded_Y)

for i in range(3):
    _X = X
    model = Sequential()
    if i == 0:
        model.add(Dense(60, init='normal', activation='relu'))
    if i == 1:
        _X = _X[:, 0:30]
        model.add(Dense(30, init='normal', activation='relu'))
    if i == 2:
        model.add(Dense(60, init='normal', activation='relu'))
        model.add(Dense(15, init='normal', activation='relu'))
        model.add(Dense(1, init='normal', activation='sigmoid'))
        model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])

    H = model.fit(_X, encoded_Y, epochs=100, batch_size=10,
validation_split=0.1, verbose=False)
    print(model.summary())
    loss = H.history['loss']
    val_loss = H.history['val_loss']
    acc = H.history['accuracy']
    val_acc = H.history['val_accuracy']
    epochs = range(1, len(loss) + 1)

    plt.figure(None, (4.6 * 3, 4.8))
    # Построение графика ошибки
    plt.subplot(121)
    plt.plot(epochs, loss, 'r.', label='Training loss')
    plt.plot(epochs, val_loss, 'b', label='Validation loss')
    plt.title('Training and validation loss')
    plt.xlabel('Epochs')
    plt.ylabel('Loss')
    plt.legend()
```

```
# Построение графика точности
plt.subplot(122)

plt.plot(epochs, acc, 'r.', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
plt.clf()
```